

Pre-01-Logism

预先知识

数据类型及对应的位宽

- `char` : 1字节
- `int` : 4字节

部件

Wiring

- Splitter
 - Fan Out: 分支数量
 - Bit Width In: 位宽
 - Bit i: 低到高->右到左; 可以设置Bottom/Top
- Pin
 - Three-state? Yes:0/1/x ; No:0/1
- Probe
 - Radix: 设置值的基数: Binary, Octal(8), Signed Decimal(10), Unsigned Decimal(10), Hexadecimal(16)
- Tunnel
 - Label: 如果两个Tunnel的Label相同, 则等效于两个Tunnel之间有导线连接
- Pull Register
 - Pull Direction: 0/1/E. 该部件将其连接的电线拉向该属性指定的值, 当且仅当电线上该点值为浮点 (Z/x) . 特别地, 该点值若是多位宽, 则任一位都会被拉向指定值
- Clock
 - High Duration: 持续输出为1的基本时间长度
 - Low Duration: 持续输出为0的基本时间长度
 - 调用: 常用于时序电路。通过Simulate启用Clock: Tick Once, Tick Enabled, Tick Frequency
- Constant
 - Data Bits: 根据Value设定
 - Value: 十六进制
- Power
 - 简介: 将全1值发送到连接上
- Ground
 - 简介: 将全0值发送到连接上, 与Power相对
- Bit Extender
 - **Extension Type:** 0, 1, Sign, Input(元件上方多出一个单比特的输入端口:0/1)

Gates

- Buffer
 - 简介：类似于导线
- XOR Gate
 - 简介：异或<->值相异为1
 - 注意：若输入数量大于2，按位异或时若对应位1的数量超过1，那么这一位的异或结果为1
- XNOR Gate
 - 简介：同或<->值相同为1
 - 注意：继承XOR的特殊部分. XNOR只将XOR的结果取反
- Odd Parity
 - 简介：若所有输入对应位中1的数量为奇数，则输出这一位的校验结果为1
- Even Parity
 - 简介：若对应位1的数量为偶数，则这一位的校验结果为1
- Controlled Buffer
 - 简介：单比特输入引脚控制输出：1（组件相当于Buffer）；else（输出是浮动值）；E（输出为E）
- Controlled Inverter
 - 简介：看作Controlled Buffer + NOT Gate

Plexers

- Multiplexer
 - 最重要的功能：从多个输入信号中选择一个作为输出
 - En: 短边端口? 0(not): 1(en)
 - Slt: 长边端口
 - Input
 - Output
- Demultiplexer
 - **Three-state? Yes(被抛弃的端口值浮动) : No(被抛弃的端口值为0)**
当DMX连接寄存器时，时钟上升沿到来时，YES：被抛弃的寄存器保存原来的值；No：被抛弃的寄存器Clear
 - Disabled Output: 决定当使能端为0时输出为0/Floating
- Decoder
 - 最重要的功能：将二进制编码转化为相应的one-hot（独热码）
 - Slt: 选择一个输出端以输出1
 - **Three-state? Yes(被抛弃的端口值浮动) : No(被抛弃的端口值为0)**
当Decoder连接寄存器时，时钟上升沿到来时，YES：被抛弃的寄存器保存原来的值；No：被抛弃的寄存器Clear

one-hot

- 方法：使用N个状态寄存器，对N个状态进行编码
- 目的：大部分算法是基于向量空间的度量进行计算，为了使非偏序关系的变量取值不具有偏序性，且到圆点是等距的。故采用独热码（也称一位编码），将离散特征的取值扩展到了欧氏空间（在众多机器学习算法中，特征之间距离的计算或相似度的计算占有重要地位。而该计算基于欧氏空间）。编码后，每一维度的特征都可以

看作连续的特征。

- 优点：解决了分类器不好处理属性数据的问题，一定程度上扩充了特征。值只有0和1，不同的类型存储在垂直的空间。
- 缺点：当类别的数量很大时，特征空间会很大。这种情况下，一般可以用PCA来减少维度。同时one hot encoding + PCA在实际中也很有用。
- PCA: Principal Component Analysis, 将n维特征映射到k维上。这k维是在原来n维特征的基础上重构的全新的正交特征，也称“主成分”。PCA坐标轴选取见下列表。通过计算数据矩阵的协方差矩阵，后得到协方差矩阵的特征值特征向量，选择特征值最大（即方差最大）的k个特征所对应的特征向量组成的矩阵。这样就可以实现数据特征的降维。
 - 第一个新坐标轴：原始数据中方差最大方向
 - 第二个新坐标轴：与第一个坐标轴正交的平面中方差最大的方向
 - 第三个新坐标轴：与第1、2个轴正交的平面中方差最大的
 - 以此类推，得到n个坐标轴

可以得到，大部分方差都包含在前k个坐标轴中，后面的坐标轴所含方差几乎为0.于是忽略余下坐标轴，仅选取前k个含有绝大多数方差的坐标轴，实现数据特征的降维处理。

- Priority Encoder
 - Input(West): 从上到下由0开始编号
 - Output(East-Top): 输出最高'1'位的编号
 - Output(East-Bottom): 输入端有1则输出1
 - Output(North): 输入端无1则输出1
 - En(South): 使能端
- Bit Selector
 - 简介：按照Output Bits将Input分成n组，最高组不够补0。从低到高由0开始编号。Slr(South)输入期望的组别编号

Arithmetic

- Adder
 - cin(North): 要加到总和的进位值。若值浮动，假定为0
 - cout(South): overflow
- Subtractor
 - bin(North): 要减到总和的借位值。若值浮动，假定为0
 - bout(South): 借位计算了。如果这些值减去为无符号数，得到的是负数，那么输出1；否则输出0
- Multiplier
 - Input(North): 要添加到元件的进位制，和西边Input乘积相加作为结果
 - Output(East): 结果的较低的Data Bits位
 - Output(South): 结果的较高的Data Bits位
- Divider
 - upper(North): 被除数的较高的Data Bits位
 - Input(West-Top): 被除数的较低的Data Bits位
 - Output(East): 商
 - rem(South): 余数
- Negator
 - 简介：求输入数据的补码（取反加一）

- Comparator
 - Numeric Type: 2's Complement / Unsigned. **使用时一定要明确数据类型**
- Shifter
- Bit Adder
 - 简介: 计算输入中有多少位是1, 并输出'1'位的数量
- Bit Finder
 - 简介: 寻找索引位置最高/最低的1/0, Output(East)表示索引编号, Output(South)表示是否找到

Memory

- RAM
 - 数据接口: Separate load and store ports
- ROM
 - 一次性将所有信息导入, 此后不可再更改

RAM和ROM的数据导入方法:

1. 手动导入

2. 文件导入: 在数据文件头增加如下一行字样 `v2.0 raw`; 输入数据要和存储器的数据位宽相匹配, 否则会出现数据截断

- Register
 - Trigger (触发模式)
 - **Enable端: 写使能, 可以使寄存器失效**

组合电路

卡诺图

- 基本构图方式
 - 把输入信号尽可能分成均匀的2组 (一组n个信号, 一组m个信号)
 - 构造一个 $2^n \times 2^m$ 矩阵
 - 组合值排布的规则: 相邻的2个组合之间只有1个值的变化
- 化简规则: 相邻的'1'位成一组, 观察出现互补的字母, 便可消掉
- 使用的一般性规则
 - 圈尽量少
 - 每个圈必须是矩形. 组内有效单元的个数必须是2的整数幂; 否则不能成为1组
 - 应该选择尽可能大的组
 - 圈可以环绕卡诺图的边界. **注意4角的单元**: 4角元素成为组合消掉
 - 避免出现单个单元, 不过如果出现这种情况会有多种选择

加法器

- 1位加法器
- 用N个1位加法器构造N位加法器
 - 串行加法器: 行波进位加法器
 - 实现: 把 C_i 与 C_{i-1} 连接, 形成进位链. 特别地, $C_0=0 \rightarrow$ 只需把 C_0 接地
 - 优点: 构造最简

- 缺点：耗时长。必须经过N个延迟才能产生结果

复用器 (MUX)

- Look Up(查找表): 只使用1个 2^{n-1} 的复用器任何N输入逻辑函数。方法: 将一个变量、0和1作为复用器的数据输入

译码器 (Decoder)

- 实现 (e.g. 2:4 Decoder)

| A_1 | A_0 | Y_3 | Y_2 | Y_1 | Y_0 |
|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

译码器的每一个输入都是一个最小项。因此可以使用与门、或门和非门实现译码器。

- 译码器逻辑
 - 和或门组合实现逻辑函数: 译码器可以和或门组合来实现逻辑函数（异或等）。因为译码器的每一个输出都是一个最小项，可实现的逻辑函数将以所有最小项的或来实现
 - 译码器参加的逻辑电路，很容易将函数表示成真值表或标准的与或式

时序

延迟

- 传播延迟: 从输入改变到一个或多个输出达到它们的最终值所经历的最长时间; 关键路径上所有元件的传播延迟之和
- 最小延迟: 从输入改变到任何一个输出开始改变所经历的最短时间; 最短路径上所有元件的最小延迟之和

毛刺

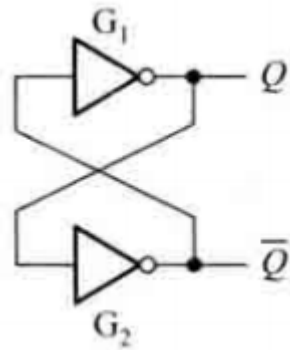
- 时机1: 信号的变化穿过卡诺图的两个主蕴涵项的边缘。解决: 可以增加多余的蕴涵项来盖住这些边缘
- 时机2: 多个输入上的同时变化。无法通过增加硬件来避免

时序电路

锁存器

- 双稳态电路

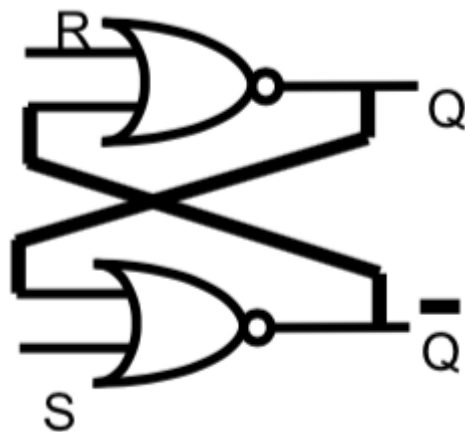
- 逻辑图



- RS锁存器

- 或非门锁存器

- 逻辑图

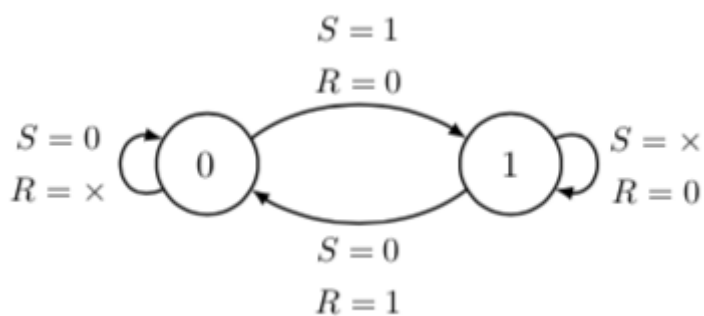


- R和S值相异时，S=1时置1，R=1时置0
- RS=0时，输出不变
- S=R=1时，有 $Q=\sim Q=0$. 违背定义 $Q=\sim Q$. 故人为规定 $SR \neq 1$
- 如何确定S和R的位置？先确定Q，则输出Q的或非门的输入为R

- 与非门锁存器

- 与或非门大致相同。不同点：S和R的位置交换，或非门低电平有效，与非门高电平有效

- 状态转移图



- RS锁存器的缺点

1. 输出不稳定：输入一旦变化，输出立刻变化。然而计算机中数据大多以先后顺序到达，那么输出会混乱。
2. 限制多，容错率低。与非门中规定 $SR \neq 1$ ，或非门规定 $S+R \neq 0$ 。

- D锁存器

- 分析

- 根据RS锁存器缺点1，引入CLK实现同步输入。
- 根据RS锁存器缺点2，将R和S用 $\sim D$ 和D表示。

- D寄存器

- 增强D寄存器的供能

寄存器相关概念

- 时钟沿
- 复位信号

- 同步复位

复位信号只有在时钟上升沿到来时，才有效。换言之，同步复位操作永远发生在时钟上升沿。即使复位信号提前到来，也无法立刻实现复位操作。

- 异步复位：

无论时钟沿是否到来，只要复位信号有效，就对系统进行复位

- 关键路径

加快时序电路的运行速度的重要手段：提高时钟频率。

然而时钟速率不能无限增长，其中一个重要的限制是寄存器之间组合逻辑电路的关键路径延迟。

有限状态机（FSM）

- 状态转移图
- 状态编码
 - 二进制编码
 - 独热码（one-hot）
 - 有效提高电路的速度和可靠性，但资源消耗较大
- 状态机类型
 - Moore：只取决于系统的状态
 - Mealy：取决于输入和系统的状态
- 状态机的分解：将复杂的状态机分解为多个互相作用的更简单的状态机
- 开发步骤
 1. 规划状态总数
 - 不能疏漏
 2. 构造状态图（刻画状态转化）
 - 每个状态的转移条件必须完备
 3. 根据{Input, S0, S1}构造每个寄存器的D输入信号的门电路
方法：真值表->乘积项表达式->最简表达式->最简门电路
 4. 根据设计需求决定当前状态输出
- 设计要点
 - 初始状态：确保系统在上电后处于确定状态
 - 复位信号：将状态机强制初始化为初始状态

仿真与调试

Simulation

- 在仿真前选中Simulation Enabled
- Step Simulation
- 子电路的调试方法

- 使用Poke Tool，双击进入想要调试的子电路
- 进入树图双击目标子电路
- 使用Simulate中的go in to state/go out to state选项，在父子电路间切换

设计指南

- Label
- 使用适当的MUX
- 使用Bit Extender扩展信号：而非使用Spiltter和Constant进行操作
- XOR Gate选择信号和其取反结果
- 使用常量时多思考是否能用其他门电路优化
- 全面测试
- 将外部库文件和电路一同上交：不要拥有对外部文件绝对路径的依赖。可以将全部文件复制到一个空文件夹上交

杂谈

Logging

- 简介：记录仿真过程中各个部件的状态及数据
- 打开方式：Simulate-Logging
- 注意事项
 - 检测前打开Simulate Enabled
 - Label:使Logging更直观

寄存器初始化

Comparator

Counter

1位计数器(Stay at value)（只在第一个时钟周期输出0，此后输出1），接到MUX的Slit端口，